

IN THE SPECIFICATION:

Please delete the paragraph starting on page 1, line 7 and replace with the following:

B1 The present invention relates generally to an improved data processing system, and in particular to a method and apparatus for managing graphic data. Still more particularly, the present invention relates to a method and apparatus for managing texture memory in a data processing system.

Please delete the paragraph starting on page 10, line 22 and replace with the following:

B2 Each texture manager, texture manager 300 and texture manager 302, allocates and frees texture memory through command to miniport 304. This is performed by each texture manager as if each one owned all of the texture manager resources. The actual allocation and free of texture memory is performed by miniport 304. Each texture manager keeps track of its allocations for all textures. If miniport 304 fails to allocate a texture memory as requested, the texture manager boots the least active texture to make room for the new allocations. After freeing, miniport 304 is again called to allocate texture memory. The process continues until the allocation succeeds or no more textures are present to be freed. If no more ~~texture~~ textures are present, texture manager 302 returns a failure. On the other hand, texture manager 300 will free up memory by removing textures handled by texture manager 302. In these examples, texture manager 300 has a higher priority over texture manager 302 and is allowed to remove textures handled by texture manager 302.

Please delete the paragraph starting on page 11, line 20 and replace with the following:

B3 The process begins by receiving a request to store a texture in texture memory (step 400). A texture is reloaded as needed (step 402). Texture memory is then allocated

B3 to the current texture in the request (step 404). In this example, the allocation occurs through a call made by the texture manager to a miniport or other memory allocation mechanism. A determination is made as to whether the allocation was successful (step 406). In this example, allocations are done on a per mipmap basis. A texture object consists of one or more mipmaps starting from level 0 and the allocation always begins with mipmap 0, then 1, and so on. A mipmap is a reduced resolution version of a texture map, used to texture a geometric primitive whose screen resolution differs from the resolution of the source texture map.
